REMARKS

Claims 1, 5 through 8, 11 and 15 through 18 and 21 are pending in this application. Claims 1, 8, 11 and 21 are currently amended. Claim 22 is canceled herein.

Claim Rejections under 35 U.S.C. §112

The Office Action rejected to claim 11 under 35 U.S.C. 112, second paragraph and to claims 15 through 17 as dependent on claim 11. The Office Action quoted a phrase from claim 11 without further explanation of the rejection. This phrase in claim 11 has been amended to correct grammatical errors.

Claim Rejections under 35 U.S.C. §102

Claims 8 and 18 are rejected under 35 U.S.C. 102(e) as being anticipated by U.S. Publication No. 20050259597 to Benedetto et al. (the Benedetto reference). A claim is anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described in a single prior art reference. The identical invention must be shown in as complete detail as contained in the claim. See M.P.E.P. 2131. Applicants respectfully traverse this rejection of the claims because the Office Action has failed to prove that the Benedetto reference discloses each element of the claims.

Independent Claim 8

The Office Action has failed to prove that the Benedetto reference discloses the elements of claim 8, *inter alia*, of, "in each of the first and second customer edge bridge of the at least one customer LAN segment having a multi-homed connection to the provider network: determining whether a topology change in the customer LAN segment affects paths of data units through the provider network in response to a status change in a blocked link between one of the first and second customer edge bridges and another edge bridge in the at least one customer LAN segment; when a topology change affects paths of data units through the provider network, transmitting flagged topology change notifications (TCNs) wherein the flagged TCNs include a set snooping flag bit that indicates the blocked link is affected by the TCN; when a topology change does not affect paths of data units through the provider network, transmitting unflagged

topology change notifications (TCNs), wherein the unflagged TCNs include a snooping flag bit that is not set; and in each of the provider edge bridges coupled to a customer LAN segment: receiving topology change notifications (TCNs) from the customer network; in response to receiving a flagged TCN, flushing an address memory file associating end host addresses with ports of the provider edge bridge; and in response to receiving an unflagged TCN, passing the TCN without flushing an address memory file." The specification of corresponding US Published Application No. 20040174828 states in paragraphs 45 through 47:

> "[0045] FIG. 5 illustrates an example of a dual-homed network where both CEs are active, with a blocked CB2-CE1 connection. With the blocked CB2-CE1 connection, both the CE1-PE1 and CE2-PE2 are single-homed, and TCNs received by the PEs can be passed through the provider network 12 without generating an unlearn command. If only the method of FIG. 4 was applied, all TCNs from CE1 would be generate an unlearn command, because it is coupled to a blocked link (CB2-CE1). In fact, only TCNs related to the reactivation of the blocked CB2-CE1 link need to be snooped; all others can be passed without initiating unlearning in the provider domain.
>
> [0046] FIG. 6 illustrates an example of an alternative method. In this case, an active CE with a blocked connection is responsible for setting a flag bit (a "snooping bit") in a BDPU, indicating whether the TCN is due to a change in status of the blocked link. This flagging need only happen at the CE and, therefore, existing bridges need not generally support it.
>
> [0047] In step 90, a customer TCN is received by the PE. In step 92, if the snooping bit is set, indicating that the blocked link is affected by the TCN, then the TCN initiates an unlearn command in step 94. Otherwise, the TCN is passed without generating an unlearn command in step 96."

Figures 5 and 6 of the specification are reproduced below.

Patent Application Publication Sep. 9, 2004 Sheet 10 of 11 US 2004/0174828 A1

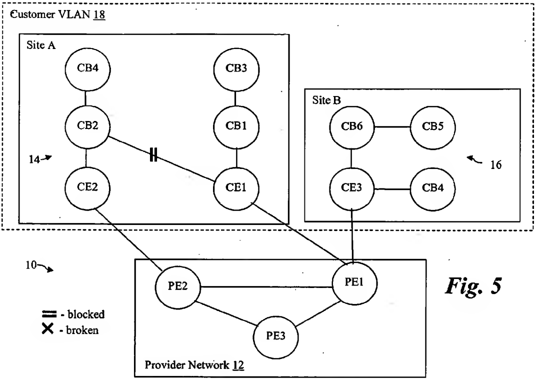
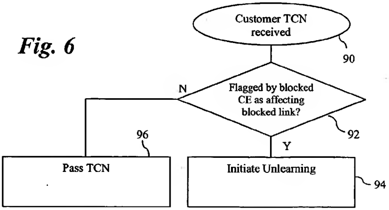


Fig. 5

Fig. 6



As described in the specification, in an embodiment, an active customer edge bridge connected to the provider network has a blocked connection. It is responsible for setting a flag bit (a "snooping bit") in a BDPU, indicating whether the TCN is due to a change in status of the

blocked link. When the snooping bit is set in the TCN, indicating that the blocked link is affected by the TCN, then the PE initiates an unlearn command of the MAC address memory file in response to the TCN. Otherwise, the TCN is passed by the PE without generating an unlearn command.

The Office Action cites Figure 2 and paragraphs 19 and 108 of the Benedetto reference for disclosing the elements of the claims. The Benedetto reference states in paragraph 19 that:

> [0019] As ports transition between the blocked and forwarding states, entities may appear to move from one port to another. To prevent bridges from distributing messages based upon incorrect address information, bridges quickly age-out and discard the "old" information in their filtering databases. More specifically, upon detection of a change in the active topology, a bridge begins transmitting Topology Change Notification Protocol Data Unit (TCN-PDU) messages on its root port. The format of the TCN-PDU message is well known (see IEEE 802.1D standard) and, thus, will not be described herein. A bridge receiving a TCN-PDU message sends a TCN-PDU of its own from its root port and sets the TCA flag 112 in BPDUs that it sends on the port from which the TCN-PDU was received, thereby acknowledging receipt of the TCN-PDU. By having each bridge send TCN-PDUs from its root port, the TCN-PDU is effectively propagated hop-by-hop from the original bridge up to the root. The root confirms receipt of the TCN-PDU by setting the TC flag 114 in the BPDUs that it subsequently transmits for a period of time. Other bridges, receiving these BPDUs, note that the TC flag 114 has been set, thereby alerting them to the change in the active topology. In response, bridges significantly reduce the aging time associated with their filtering databases which, as described above, contain destination information corresponding to the entities within the network. Specifically, bridges replace the default aging time of five minutes with the forwarding delay time, which by default is fifteen seconds. Information contained in the filtering databases is thus quickly discarded.

The Benedetto reference only describes that upon detection of a change in the active topology, a bridge begins transmitting Topology Change Notification Protocol Data Unit (TCN-PDU) messages on its root port. Furthermore, the Benedetto reference only describes that a TCA flag 112 in BPDUs is set thereby ***acknowledging receipt*** of the TCN-PDU. In general, a bridge acknowledges the reception of a TCN BPDU by setting the TCA flag in its next configuration BPDU. The Benedetto reference nowhere describes determining whether a topology change in the customer LAN segment affects paths of data units through the provider network in response to a status change in a blocked link between one of the first and second customer edge bridges and another edge bridge in the at least one customer LAN segment; when a topology change affects paths of data units through the provider network, transmitting flagged topology change notifications (TCNs) wherein the flagged TCNs include a set snooping flag bit that indicates the blocked link is affected by the TCN.

In conclusion, the Benedetto reference fails to disclose each element of the independent claim 8 and thus fails to anticipate claim 8 under 35 U.S.C. 102(e).

Independent Claim 18 and dependent claims 19 and 20

For similar reasons as stated with respect to claim 8, the Office Action has failed to prove that the Benedetto reference discloses the elements of claim 18.

Claim Rejections under 35 U.S.C. §103

The Office Action rejected claims 1, 5-7, 11, 15-17, and 21 under 35 U.S.C. 103 as being unpatentable over the Benedetto reference in view of U.S. Patent No. 7,277,399 to Hughes (the Hughes reference). Applicants respectfully traverse the rejection of these claims under 35 U.S.C. §103(a) because the Office Action has failed to provide a prima facie case of obviousness of the claims in view of the cited references for the reasons stated below.

Independent Claim 1 and dependent claims 5 through 7, 21 and 22

The Office Action has failed to provide a prima facie case of obviousness of claim 1 because the combination of the Benedetto reference and the Hughes reference teaches away from the elements in claim 1, *inter alia*, of, "in response to receiving a TCN from the customer

network, monitoring end host media access control (MAC) addresses in data units received from the customer network for a predetermined time period; determining whether a topology change has occurred in one or more of the customer virtual LAN segments that affects paths of data units through the provider network by: monitoring whether a predetermined number of new end host MAC addresses of data units received from the customer network in the predetermined time period are found in a MAC address memory file, wherein the MAC address memory file associates end host MAC addresses with ports of the provider edge bridge; and monitoring whether a contradiction occurs between an end host MAC address of a data unit received from the customer network and the MAC address memory file; in response to determining a topology change in one or more of the customer virtual LAN segments do not affect paths of data units through the provider network, storing a new address in the MAC address memory file without flushing the MAC address memory file." The specification of corresponding US Published Application No. 20040174828 states in paragraphs 11 and 12 that:

> "[0011] Topology changes in the Customer VLAN can also require changes in the MAC address tables in the bridges of the provider network. A previous proposal allows snooping on all TCNs generated within the customer domain, and taking action indiscriminately. Each time a TCN is generated, the provider domain unlearns (flushes the MAC table of each bridge) and re-learns all the addresses. Re-learning the MAC address at each bridge is a costly and time-consuming operation.
>
> [0012] Therefore, a need has arisen for an efficient method of taking action inside the provider domain responsive to TCNs received from the customer domain.

The specification of corresponding US Published Application No. 20040174828 states in paragraphs 33 through 37 that:

> [0033] FIG. 1b illustrates a topology change that will affect the MAC address tables of the bridges in Site A, but does not affect the MAC address table in any of the provider bridges. In FIG. 1b, the CB2-CB3 link is activated and the CE1-

> CB1 link is blocked. TCNs will be generated accordingly within the Customer VLAN 18 to indicate that changes have been made to the topology. TCNs are generated as a BPDU (Bridge Protocol Data Unit); if the TCN flag is set in a BDPU, it is interpreted as a TCN. The MAC address tables for CE1 and CB2 must be changed in response to these TCNs, but the forwarding address in the provider bridges remain valid with regard to addresses X and Y. Therefore, unlearning address in the provider domain due these TCNs would be wasteful.
>
> . . .
>
> [0037] The flowchart of FIG. 2a determines whether a TCN was generated for a topology change in the **customer domain** that may require unlearning/relearning operations in the **provider domain**, by checking for contradictory MAC address or new MAC addresses. However, receiving a new MAC address does not conclusively mean that the new address was received due to the topology change indicated by the received TCN. A new MAC address could also indicate that a MAC address was not previously active. Thus, an unlearning operation could be unnecessary."

The specification describes that in known solutions, each time a TCN is generated from a customer domain, a provider domain unlearns (flushes the MAC table of each bridge) and re-learns all the addresses. However, re-learning the MAC address at each bridge is a costly and time-consuming operation. An embodiment of the specification describes that a TCN generated from a customer domain may not require flushing of the MAC addresses in the provider domain when the topology change in the customer domain does not affect paths of data units through the provider domain. Thus, an embodiment determines whether the MAC addresses need to be flushed and relearned in a provider domain in response to a TCN from a customer domain. If the topology change in the customer domain does not affect paths of data units through the provider domain, then the MAC addresses in the address memory file are not flushed and relearned.

The Office Action cites Figure 2 and paragraphs 19, 92 and 113 of the Benedetto reference for disclosing these elements of the claims. The Benedetto reference states in paragraph 17 that:

> If a bridge *stops receiving BPDU messages on a given port* (indicating a possible link or device failure), it will continue to increment the respective message age value until it reaches the maximum age threshold. The bridge will then discard the stored BPDU information and proceed to re-calculate the root, root path cost and root port by transmitting BPDU messages utilizing the next best information it has. The maximum age value used within the bridged network is typically set by the root, which enters the appropriate value in the maximum age field 126 of its transmitted BPDU messages 100. Neighboring bridges similarly load this value in their BPDU messages, thereby propagating the selected value throughout the network. The default maximum age value under the IEEE standard is twenty seconds. [emphasis added]

The Benedetto reference clearly indicates in this paragraph 17 that *a bridge will discard stored BPDU information if a bridge stops receiving BPDU messages on a given port for a maximum age value threshold.* In paragraph 19, the Benedetto reference reiterates this disclosure and states that:

> To prevent bridges from distributing messages based upon incorrect address information, bridges quickly age-out and discard the "old" information in their filtering databases. More specifically, upon detection of a change in the active topology, a bridge begins transmitting Topology Change Notification Protocol Data Unit (TCN-PDU) messages on its root port. The format of the TCN-PDU message is well known (see IEEE 802.1D standard) and, thus, will not be described herein. A bridge receiving a TCN-PDU message sends a TCN-PDU of its own from its root port and sets the TCA flag 112 in BPDUs that it sends on the port from which the TCN-PDU was received, thereby acknowledging receipt of the TCN-PDU. By having each bridge send TCN-PDUs from its root port, the TCN-PDU is effectively propagated hop-by-hop from the original bridge up to the root. The root confirms receipt of the TCN-PDU by setting the TC flag 114 in the BPDUs that it subsequently transmits for a period of time. Other bridges,

> receiving these BPDUs, note that the TC flag 114 has been set, thereby alerting them to the change in the active topology. In response, bridges significantly reduce the aging time associated with their filtering databases which, as described above, contain destination information corresponding to the entities within the network. Specifically, bridges replace the default aging time of five minutes with the forwarding delay time, which by default is fifteen seconds. Information contained in the filtering databases is thus quickly discarded.

Thus, in response to a TCN-PDU, the Benedetto reference discloses that the filtering databases always flush the filtering databases within fifteen seconds when a BPDU includes a TCN, e.g. with the TC flag set. Information contained in the filtering databases is thus always discarded as described in the Benedetto reference within the aging time associated with their filtering databases. This teaches the same problem described in paragraph 11 of the specification that each time a TCN is generated, the provider domain unlearns (flushes the MAC table of each bridge) and re-learns all the addresses. Re-learning the MAC address at each bridge is a costly and time-consuming operation. As such, the Benedetto reference teaches away from embodiments of claim 1 that a TCN generated from a customer domain may not require flushing of the MAC addresses in the provider domain when the topology change in the customer domain does not affect paths of data units through the provider domain.

The Hughes reference fails to add to the teachings of the Benedetto reference. The Office Action argues on page 7 that:

> "Hughes discloses in response to determining a topology change in one or more of the customer LAN segments do not affect paths of data units through the provider network, storing a new address in the address memory file without flushing the address memory file (Col.6 lines 51-67 When network topology is update, the router receives topology updates including new destination addresses. The router can select individual entries in the hardware-based route cache that require updating and updates the individual entry without flushing the hardware-based route cache). Benedetto and Hughes are analogous because they both

> pertain to data communication. It would have been obvious to one of ordinary skill in the art at the time of the invention to modify Benedetto to include storing a new address in the address memory file without flushing the address memory file as taught by Hughes in order to more efficiently update a route cache."

However, the Office Action has misinterpreted the Hughes reference. The Hughes reference relates to a hardware-based route cache in a router. The Hughes reference states that:

> "Generally, a router in a network uses routing tables to lookup a destination address to compute network routing and forward an incoming packet. Routing tables typically store millions of destination addresses for network hosts. Routing tables are periodically updated to reflect the active status of hosts in the network. When a packet arrives at the a router, the router extracts the destination information from packet header and searches the routing table for the destination route. Because the routing tables are based on conventional memory technology and contain a large amount of routing information, a search for the destination route can take significant amount of time and in some cases, may cause the router to drop the incoming packet due to certain timeout limits.
>
> One method to resolve route lookup delay is to implement a route cache. Typically, a route cache is a software-based search table. A route cache is a comparatively smaller lookup table that stores the addresses of certain selected destinations (e.g., most frequently accessed destinations, recently accessed destinations and the like). When the router receives a packet, the router first searches the route cache for routing information and if the routing information is found in the route cache, the router forwards the packet to appropriate destination. Because the route cache is smaller, the search time is significantly less than the search time of larger lookup tables. The route cache is populated based on statistical and data traffic analysis done by each router in the network. A problem with software-based route caches is that the router keeps adding destination addresses to the route cache until the length of route cache reaches a certain

> maximum limit and the search efficiency starts degrading router's performance. The router then flushes the route cache (clears all the entries in the cache) and repopulates the route cache. The flushing and repopulating of the route cache causes additional delays for packet routing. This additional delay severely affects router performance and limits the maximum allowable throughput of the router."

As described in the Hughes reference, the route cache is populated based on statistical and data traffic analysis done by each router in the network to include a comparatively smaller lookup table than the full routing tables. The Hughes reference describes that when the length of route cache reaches a certain maximum limit and the search efficiency starts degrading router's performance, the router then flushes the route cache (clears all the entries in the cache) and repopulates the route cache. Thus, the route cache is flushed when it reaches a certain maximum limit and is then repopulated based on statistical and data traffic analysis. The Hughes reference further explains that:

> "Hardware-based route cache can be configured as self-managed cache. When the hardware-based route cache is full and the router receives a packet with new destination address that requires caching, the router can replace an existing entry in the hardware-based route cache. The router may employ multiple schemes to replace certain entries in the hardware-based route cache (such as, for example, first-in-first-out (FIFO), least recently used address, least active address, oldest address, low priority address, random selection or the like). The advantage of hardware-based route cache over conventional search methods is that individual entry in the hardware-based route cache can be replaced without flushing the cache. Frequent arrival of packets with non-cached destinations when the cache is full, may require frequent flushing and rebuilding of a memory-based search table. However, in a hardware-based route cache, individual entries can be replaced without the flushing and rebuilding of the cache.
>
> In one embodiment of the present invention, when network topology is updated (i.e., e.g., network addresses of destinations are changed, servers are removed

> from service or the like), the router receives topology updates including new destination addresses. The router then updates the routing tables accordingly. Once the routing tables are updated, the router can determine whether to update the hardware-based route cache, if the hardware-based route cache requires updating, the respective entry in the hardware-based route cache can be repopulated to reflect the change in the routing table. In one embodiment, the router can flush the hardware-based route cache and rebuild the hardware-based route cache to reflect the updated network topology. In an embodiment, the router can select the individual entries in the hardware-based route cache that require updating and updates the individual entry without flushing the hardware-based route cache."

As described above, when the router receives a topology update, it updates the routing tables accordingly. The hardware based routing cache may be updated to reflect changes in the routing tables or the router can flush the hardware-based route cache and rebuild the hardware-based route cache to reflect the updated network topology in the routing tables. Thus, the Hughes reference is only describing a hardware-based route cache in a router and how to update the route cache to reflect changes to the routing tables. There is no description or discussion of that a TCN generated from a customer domain may not require flushing of the MAC addresses in a MAC memory address table in a provider domain when the topology change in the customer domain does not affect paths of data units through the provider domain.

In conclusion, the Office Action has failed to show how the Benedetto reference and the Hughes reference discloses or makes obvious the elements of claim 1 under 35 U.S.C. §103. Instead, the Benedetto references teaches the same problem that each time a TCN is generated, the provider domain unlearns (flushes the MAC table of each bridge) and re-learns all the addresses. The Hughes reference is not even related to VLANs and MAC addressing or how to update a MAC address table or determining when a topology change in the customer domain does not affect paths of data units through the provider domain. As such, the Office Action has failed to prove that the combination of the Benedetto reference and the Hughes reference discloses or makes obvious the elements of claim 1.

Since the references teach away from the elements of claim 1 or are unrelated to claim 1, it seems that keywords in claim 1 and teachings of the specification were used to select the cited references and piecemeal together the rejection under 35 U.S.C. 103. "The court must be ever alert not to read obviousness into an invention on the basis of the applicant's own statements; that is, we must view the prior art without reading into that art appellant's teachings." Application of Nomiya, 184 U.S.P.Q. 607, 612 (Cust. & Pat.App. 1975). The citation of the specification's own teachings to argue obviousness over prior art is improper. In re Dembiczak, 175 F.3d 994, 999, (criticizing hindsight syndrome wherein that which only the inventor taught is used against the teacher).

Claims 5 through 7, 21 and 22 add further patentable matter to Claim 1 and thus are further differentiated and patentable under 35 U.S.C. §102 over the cited references.

<u>Independent Claim 11 and dependent claims 15 through 17</u>

For similar reasons as stated with respect to claim 1, the Office Action has failed to prove that the combination of the Benedetto reference and the Hughes reference discloses or makes obvious the elements of claim 11. Claims 15 through 17 add further patentable matter to Claim 11 and thus are further differentiated and patentable under 35 U.S.C. §103 over the combination of the Benedetto reference and the Hughes reference.

CONCLUSION

For the above reasons, the foregoing amendment places the Application in condition for allowance. Therefore, it is respectfully requested that the rejection of the claims be withdrawn and full allowance granted. Should the Examiner have any further comments or suggestions, please contact Jessica Smith at (972) 240-5324.

Respectfully submitted,
GARLICK HARRISON & MARKISON

Dated: November 30, 2009

/Jessica Smith/

Jessica W. Smith
Reg. No. 39,884

Garlick Harrison & Markison
P. O. Box 160727
Austin, TX 78716-0727
Phone: (972) 240-5324
Fax: (888) 456-7824